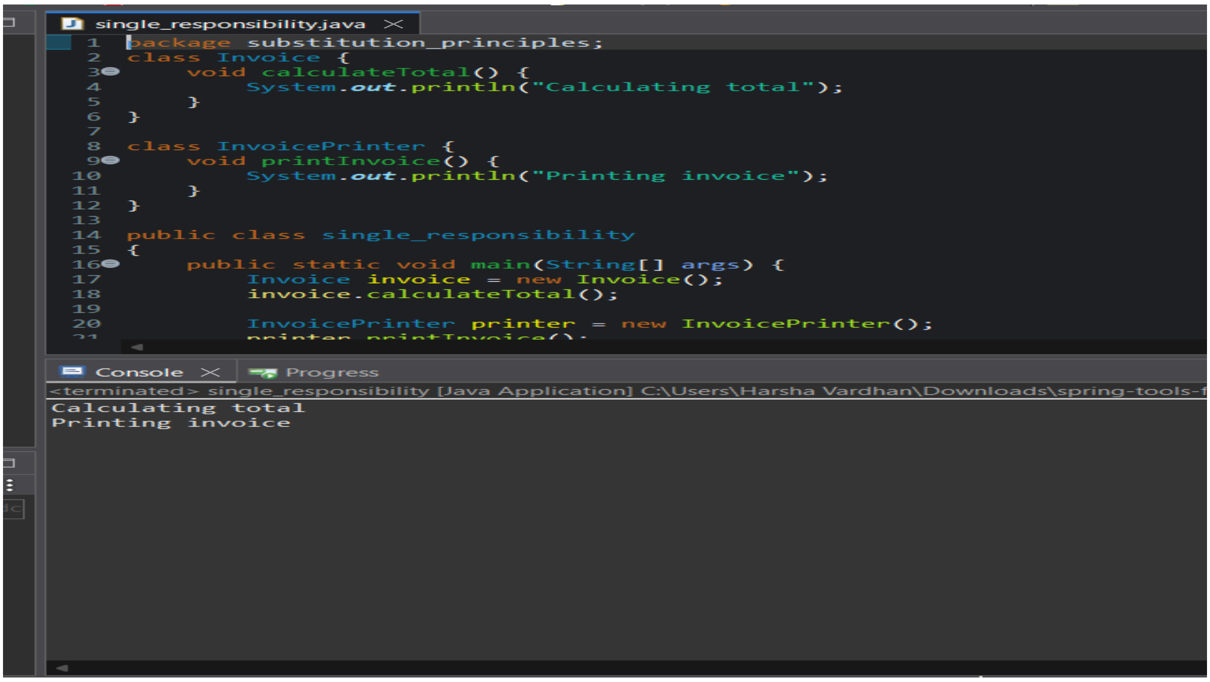
Solid_Principles_proofs

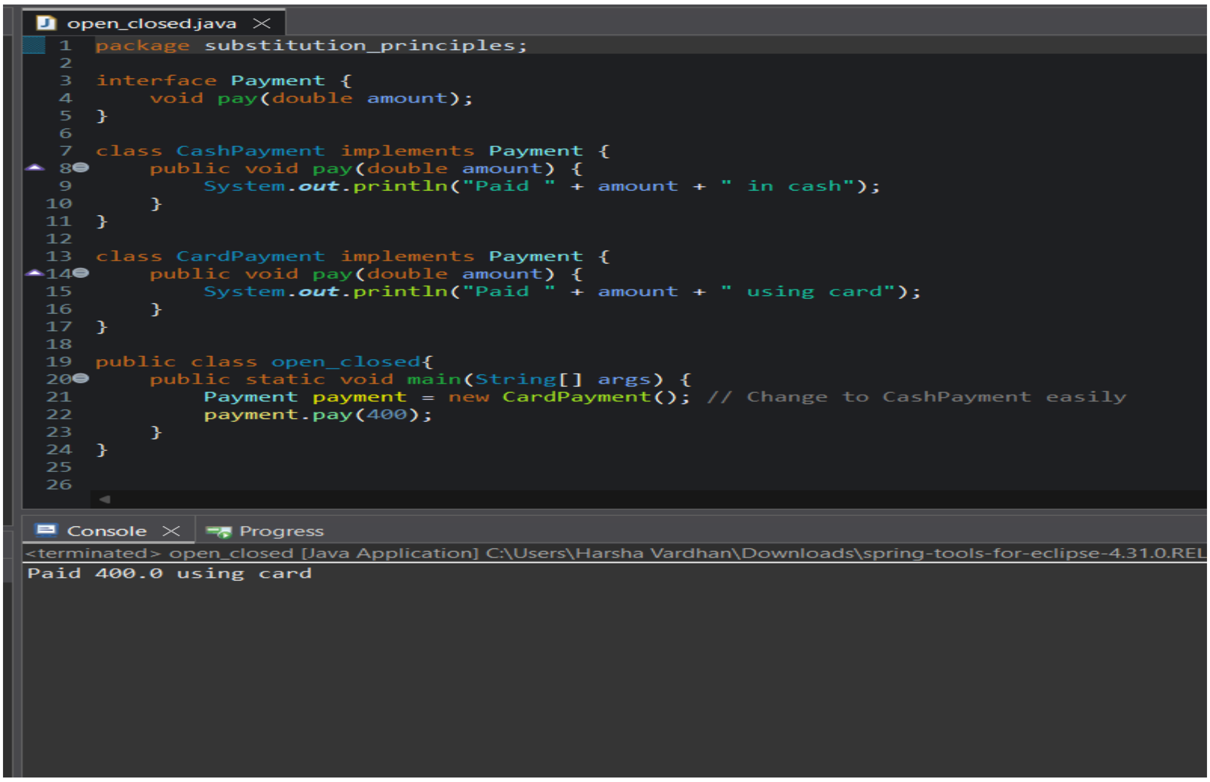
1.Single_Responsibility



```java
package substitution_principles;
class Invoice {
	void calculateTotal() {
		System.out.println("Calculating total");
	}
}

class InvoicePrinter {
	void printInvoice() {
		System.out.println("Printing invoice");
	}
}

public class single_responsibility
{
	public static void main(String[] args) {
		Invoice invoice = new Invoice();
		invoice.calculateTotal();

		InvoicePrinter printer = new InvoicePrinter();
		printer.printInvoice();
```

```
<terminated> single_responsibility [Java Application] C:\Users\Harsha Vardhan\Downloads\spring-tools-f
Calculating total
Printing invoice
```

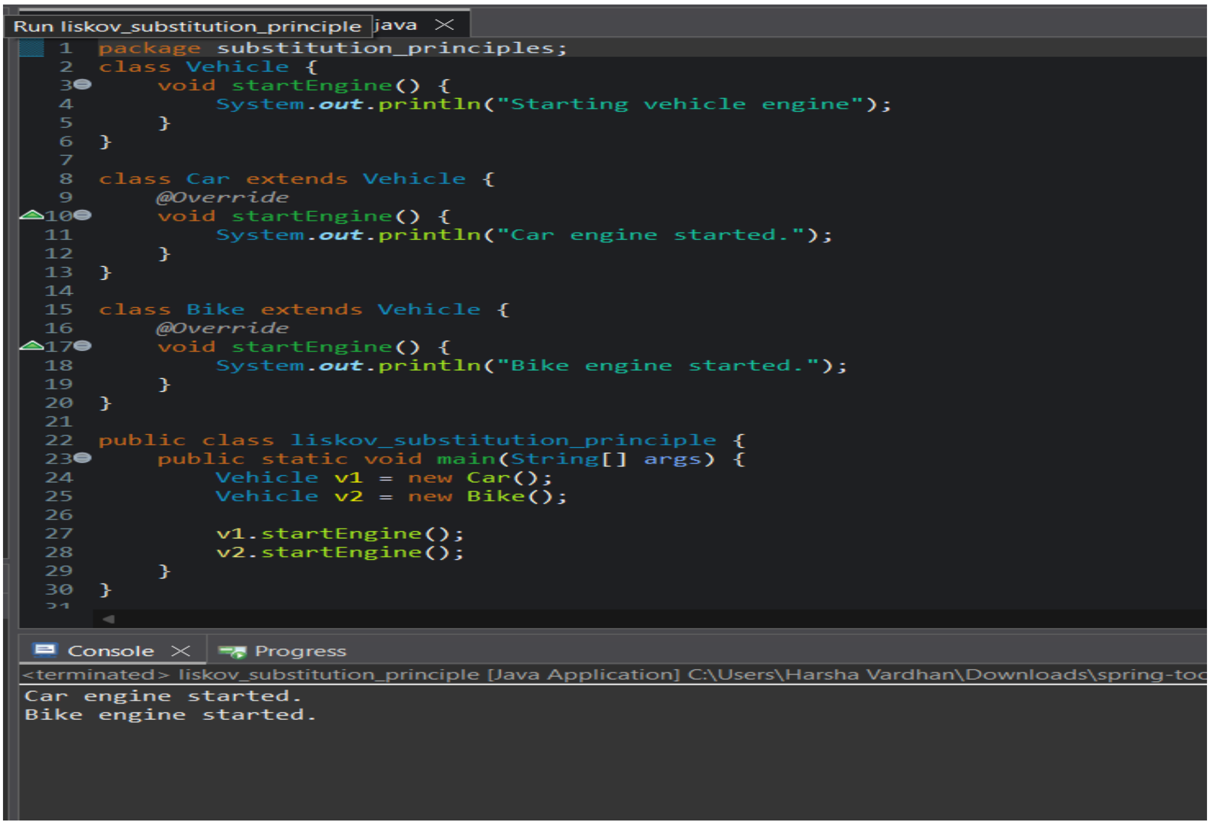
2.Open_Closed



```java
package substitution_principles;

interface Payment {
	void pay(double amount);
}

class CashPayment implements Payment {
	public void pay(double amount) {
		System.out.println("Paid " + amount + " in cash");
	}
}

class CardPayment implements Payment {
	public void pay(double amount) {
		System.out.println("Paid " + amount + " using card");
	}
}

public class open_closed{
	public static void main(String[] args) {
		Payment payment = new CardPayment(); // Change to CashPayment easily
		payment.pay(400);
	}
}
```

```
<terminated> open_closed [Java Application] C:\Users\Harsha Vardhan\Downloads\spring-tools-for-eclipse-4.31.0.REL
Paid 400.0 using card
```

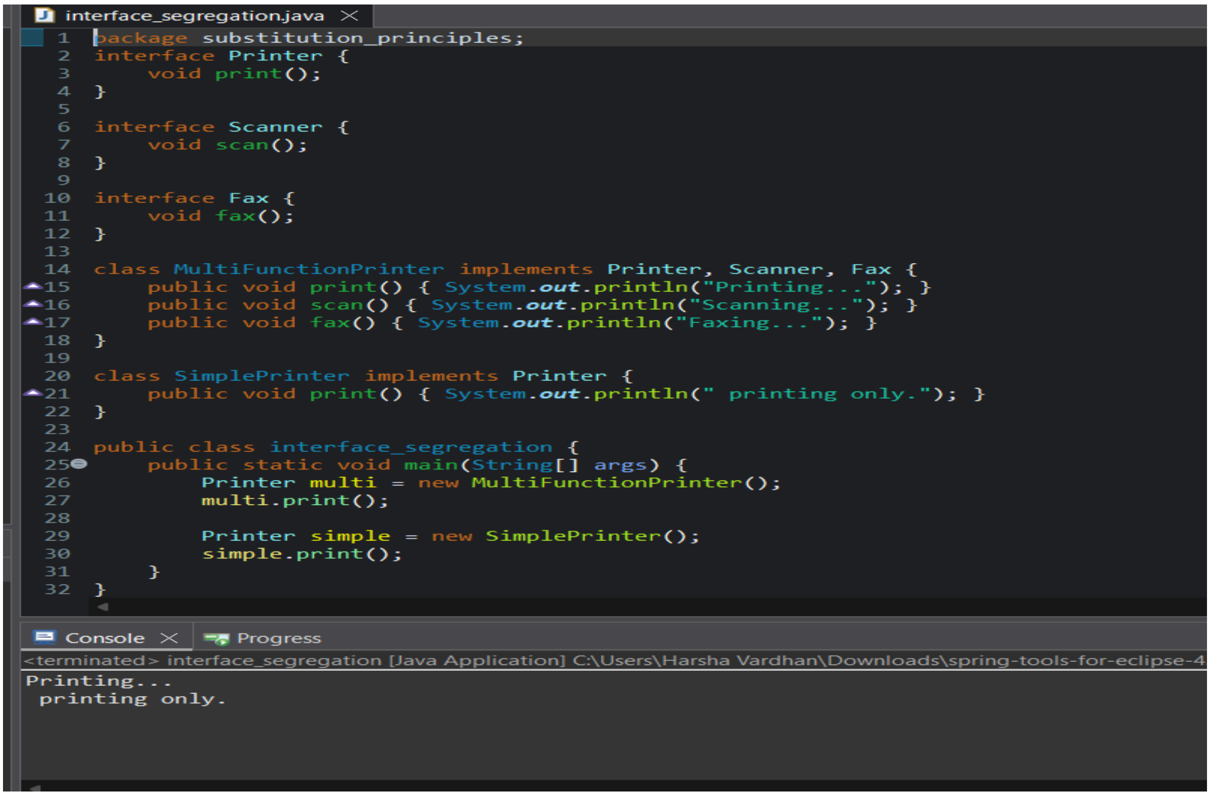
3.Liskov_Substitution



```java
package substitution_principles;
class Vehicle {
    void startEngine() {
        System.out.println("Starting vehicle engine");
    }
}

class Car extends Vehicle {
    @Override
    void startEngine() {
        System.out.println("Car engine started.");
    }
}

class Bike extends Vehicle {
    @Override
    void startEngine() {
        System.out.println("Bike engine started.");
    }
}

public class liskov_substitution_principle {
    public static void main(String[] args) {
        Vehicle v1 = new Car();
        Vehicle v2 = new Bike();

        v1.startEngine();
        v2.startEngine();
    }
}
```

```
<terminated> liskov_substitution_principle [Java Application] C:\Users\Harsha Vardhan\Downloads\spring-toc
Car engine started.
Bike engine started.
```

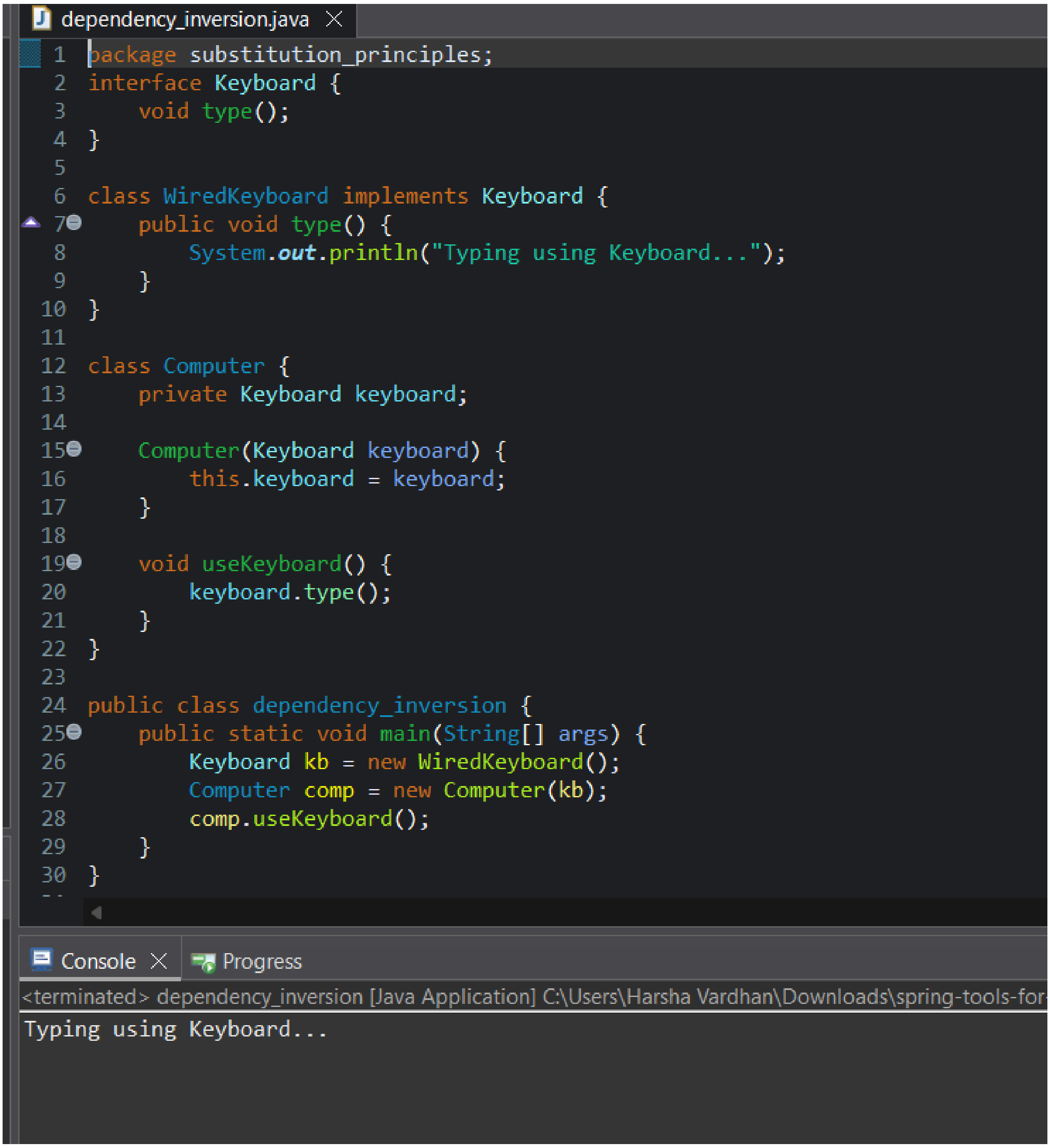
4.Interface _Segregation



```java
package substitution_principles;
interface Printer {
    void print();
}

interface Scanner {
    void scan();
}

interface Fax {
    void fax();
}

class MultiFunctionPrinter implements Printer, Scanner, Fax {
    public void print() { System.out.println("Printing..."); }
    public void scan() { System.out.println("Scanning..."); }
    public void fax() { System.out.println("Faxing..."); }
}

class SimplePrinter implements Printer {
    public void print() { System.out.println(" printing only."); }
}

public class interface_segregation {
    public static void main(String[] args) {
        Printer multi = new MultiFunctionPrinter();
        multi.print();

        Printer simple = new SimplePrinter();
        simple.print();
    }
}
```

```
<terminated> interface_segregation [Java Application] C:\Users\Harsha Vardhan\Downloads\spring-tools-for-eclipse-4
Printing...
 printing only.
```

5.Dependency _Inversion



dependency_inversion.java

```java
package substitution_principles;
interface Keyboard {
    void type();
}

class WiredKeyboard implements Keyboard {
    public void type() {
        System.out.println("Typing using Keyboard...");
    }
}

class Computer {
    private Keyboard keyboard;

    Computer(Keyboard keyboard) {
        this.keyboard = keyboard;
    }

    void useKeyboard() {
        keyboard.type();
    }
}

public class dependency_inversion {
    public static void main(String[] args) {
        Keyboard kb = new WiredKeyboard();
        Computer comp = new Computer(kb);
        comp.useKeyboard();
    }
}
```

```
<terminated> dependency_inversion [Java Application] C:\Users\Harsha Vardhan\Downloads\spring-tools-for
Typing using Keyboard...
```